We claim:

1. A method for providing data security in a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, the method comprising the steps of:

detecting an I/O request to said first device driver;

determining whether said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, denying the I/O request in said first device driver, and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

2. The method of claim 1 wherein said first device driver is a file system monitor.

3. The method of claim 1 wherein the data is stored in a secure virtual file system, and wherein the step of performing the I/O request further comprises the step of implementing data security measures.

4. The method of claim 1 wherein the data is stored in encrypted form, and wherein the step of performing the I/O request further comprises the step of decrypting the data.

5. The method of claim 1 wherein the step of performing the I/O request further comprises the step of checking the data for viruses.

6. The method of claim 1 wherein the step of determining whether said first device driver is functionally uppermost in the layered plurality of device drivers further comprises the steps of:

determining whether said first device driver has been previously called;

if said first device driver has not been previously called, detecting an initial calling module address, storing said initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, detecting a second calling module address, comparing said second calling module address to the initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers only if the initial calling module address matches the second calling module address.

7. The method of claim 1 wherein the step of denying the I/O request in the secure first device driver comprises the steps of:

setting a first device driver shutdown flag; and

initiating a re-hook process.

8. The method of claim 1 further comprising, after the step of detecting an I/O request to said first device driver, the steps of:

checking whether a first device driver shutdown flag is set; and

if said first device driver shutdown flag is set, omitting further steps in said first device driver, and allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

9. The method of claim 7 wherein the step of initiating a re-hook process further comprises the steps of:

counting the number of times the re-hook process has been initiated;

checking whether the number of times has reached a predetermined maximum threshold;

if the number of times has reached a predetermined maximum threshold, initiating a programmable security response;

if the number of times has not reached a predetermined maximum threshold, initiating reattachment of said first device driver functionally uppermost in the layered plurality of device drivers;

if said first device driver has been reattached functionally uppermost in the layered plurality of device drivers, unsetting said first device driver shutdown flag; and

concluding the re-hook process.

10. The method of claim 9 wherein the programmable security response comprises the step of destroying the data.

11. The method of claim 9 wherein the data is stored in a secure virtual file system, and wherein the step of destroying the data further comprises the step of destroying the secure virtual file system.

12. The method of claim 9 wherein the programmable security response comprises the step of terminating open applications.

13. The method of claim 9 wherein the programmable security response comprises the step of destroying said first device driver on the data storage device.

14. The method of claim 9 wherein the programmable security response comprises the step of halting the operation of the computer.

15. The method of claim 9 wherein the programmable security response comprises the step of causing the computer to enter a state requiring reboot.

16. A system for providing data security, the system comprising a first device driver operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, wherein said first device driver:

detects an I/O request;

determines whether said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver is functionally uppermost in the layered plurality of device drivers, performs the I/O request; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, denies the I/O request, and allows the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

17. The method of claim 16 wherein said first device driver is a file system monitor.

18. The system of claim 16 further comprising a secure virtual file system for storing the data, and wherein said first device driver performs the I/O request by implementing data security measures.

19. The system of claim 16 wherein the data is stored in encrypted form, and wherein said first device driver performs the I/O request by decrypting the data.

20. The system of claim 16 wherein said first device driver performs the I/O request by checking the data for viruses.

21. The system of claim 16 wherein said first device driver determines whether it is functionally uppermost in the layered plurality of device drivers by:

determining whether said first device driver has been previously called;

if said first device driver has not been previously called, detecting an initial calling module address, storing said initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver has been previously called, detecting a second calling module address, comparing said second calling module address to the initial calling module address, and concluding that said first device driver is functionally uppermost in the layered plurality of device drivers only if the initial calling module address matches the second calling module address.

22. The system of claim 16 further comprising a first device driver shutdown flag and a re-hook system, wherein said first device driver denies the I/O request by setting a first device driver shutdown flag and calling the re-hook system.

23. The system of claim 16 further comprising a first device driver shutdown flag, wherein, after said first device driver detects an I/O request, said first device driver:

checks whether a first device driver shutdown flag is set; and

if said first device driver shutdown flag is set, omits further steps in said first device driver, and allows the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.

24. The system of claim 22 wherein the re-hook system further comprises a counter that counts the number of times the re-hook system has been initiated, and wherein the re-hook system:

checks whether the number of times has reached a predetermined maximum threshold;

if the number of times has reached a predetermined maximum threshold, initiates a programmable security response;

if the number of times has not reached a predetermined maximum threshold, initiates

reattachment of said first device driver functionally uppermost in the layered plurality of device drivers; and

if said first device driver has been reattached functionally uppermost in the layered plurality of device drivers, unsets said first device driver shutdown flag.

25. The system of claim 24 wherein the programmable security response destroys the data.

26. The system of claim 24 further comprising a secure virtual file system for storing the data, and wherein the programmable security response destroys the data and destroys the secure virtual file system.

27. The system of claim 24 wherein the programmable security response terminates open applications.

28. The system of claim 24 wherein the programmable security response destroys said first device driver on the data storage device.

29. The system of claim 24 wherein the programmable security response halts the operation of the computer.

30. The system of claim 24 wherein the programmable security response causes the computer to enter a state requiring reboot.

31. A machine-readable medium comprising secured data and a first device driver program for providing data security when operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, said first device driver program comprising:

computer-implemented instructions for detecting an I/O request to said first device driver;

computer-implemented instructions for determining whether said first device driver is functionally uppermost in the layered plurality of device drivers;

computer-implemented instructions for performing the I/O request in said first device driver if said first device driver is functionally uppermost in the layered plurality of device drivers; and

computer-implemented instructions for denying the I/O request in said first device driver if said first device driver is not functionally uppermost in the layered plurality of device drivers, and for allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers if said first device driver is not functionally uppermost in the layered plurality of device drivers.

32. A computer-implemented first device driver for providing data security when operably installed in a computer operating system having a layered plurality of device drivers for accessing data in a data storage device, said first device driver comprising:

means for detecting an I/O request to said first device driver;

means for determining whether said first device driver is functionally uppermost in the layered plurality of device drivers;

if said first device driver is functionally uppermost in the layered plurality of device drivers, means for performing the I/O request in said first device driver; and

if said first device driver is not functionally uppermost in the layered plurality of device drivers, means for denying the I/O request in said first device driver, and means for allowing the I/O request to be performed by a next lower-level device driver in the layered plurality of device drivers.